8.0 APPLICATION WINDOW DESIGN

8.1 PRIMARY AND SECONDARY TASK WINDOWS

8.1.1 Window Components

A primary task window contains a window frame with resize borders, Window menu, Minimize, and Maximize buttons, and a title bar.

Motif Only: The Window menu includes (in this order) Restore, Move, Size, Minimize, Maximize, Lower, Occupy Workspace, Occupy All Workspaces, Unoccupy Workspace, and Close options.

Windows Only: The Window menu includes (in this order) Restore, Move, Size, Minimize, Maximize, and Close options and may include Switch To and Next options.

The area inside the window frame of a primary task window contains a title and a main area. The window also includes a menu bar and may have a message bar, as shown in figure 8-1.

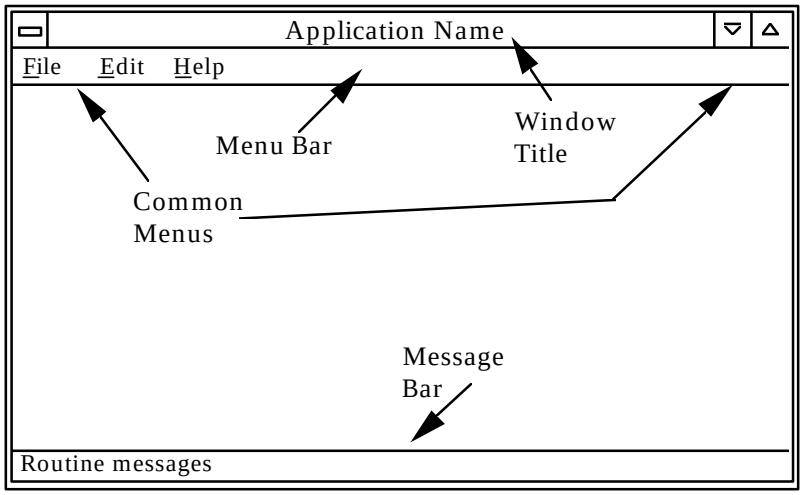


Figure 8-1. Example primary task window in Windows.

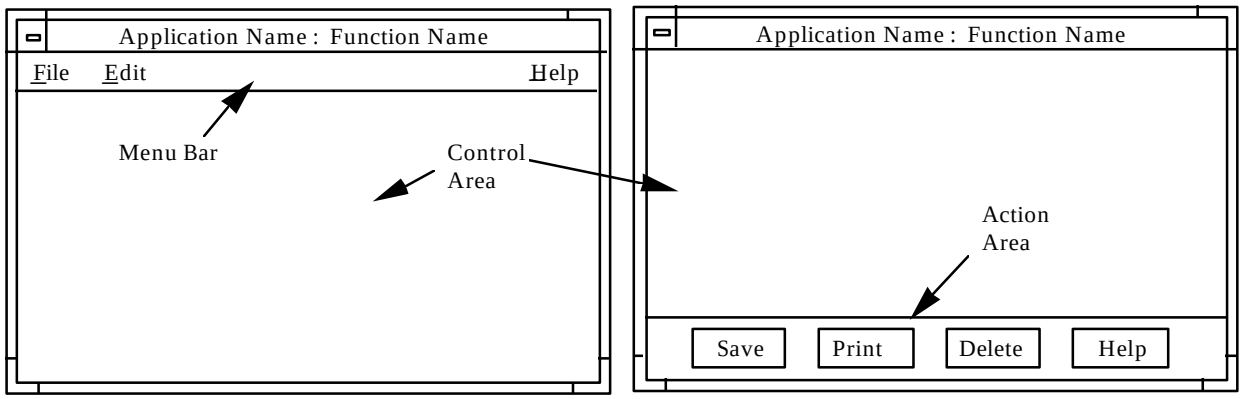
A secondary task window contains a window frame with resize borders, a Window menu button, and a title bar; the window may also contain Minimize and Maximize buttons.[1]

Motif: The Window menu in a secondary task window includes (in this order) Restore, Move, Size, Minimize (if available), Maximize (if available), Lower, and Close options.

Windows: The Window menu in a secondary task window includes (in this order) Restore, Move, Size, Minimize (if available), Maximize (if available), and Close options.

[1] Although Motif and Windows indicate that secondary windows do not have a Minimize button, the specifications presented here allow this component to be present in order to provide greater flexibility in window management (e.g., so that users have easy access to window functionality without obscuring critical information in a primary window).

The area inside the window frame of a secondary task window contains a title and a main area. The window includes either a menu bar at the top or an action area at the bottom, as shown in figure 8-2; the window may also have a message bar. The window includes a menu bar if the number of user operations is more than five and/or if access to File and Edit operations is needed. The window includes an action area if the number of user operations is five or less and/or if access is needed to application data but not to File or Edit operations.



Menubar Secondary Task Window Push button Secondary Task Window

Figure 8-2. Example secondary task windows in Motif.

8.1.2 Window Design Guidelines

8.1.2.1 Window Title

The window title is centered in the title bar and presented in mixed case, with the first letter of each word capitalized. If a file name is included in the title, it is in mixed case in Motif and in upper case letters (including any extensions) in Windows. In both cases, a placeholder name (e.g., Untitled Document 1) is used if the file has not yet been named. Specifications concerning window naming are presented in section 7.2 on application design.

Each window title in the application is unique. The window title does not contain information such as the version of an application or the full path name for a file. In addition, the window title is not used to present dynamic information such as messages to the user. If selecting a menu option causes a secondary window to be displayed, the title of the window matches or refers to the wording of the option that displayed it.

8.1.2.2 Menu Bar

If a window includes a menu bar, it appears below the title bar and contains no more than ten menu titles plus Help.

Motif Only: The menu titles begin at the left margin of the menu bar and extend rightward, with Help at the right margin of the menu bar. The titles are displayed in buttons of equal size, using the default dimensions defined by Motif, except for margin width which is 8 pixels (per Kobara).

Windows Only: The menu titles begin at the left margin of the menu bar and extend rightward, with Help as the last menu. Help is placed next to the menu that precedes it.

The space between menu titles is sufficient (at least three character widths) so that multi-word titles can be distinguished from single-word titles. Commands (e.g., push buttons) are not included in the menu bar.

8.1.2.3 Common Menus

Motif and Windows conventions concerning menu design and content are followed except as needed to provide access to application-specific functions. If any of the following common menus is used, they are ordered: File, Edit, View, Options, Window (Windows only), Help. Application-specific menus can be inserted between these common ones, except as indicated below.

File menu. The first (i.e., leftmost) menu in the menu bar contains options for users to work with the data in the window as a whole. The title of this menu is File or an application-specific term with comparable meaning.

Motif Only: If the File menu includes any of the following options, they are ordered: New, Open, Save, Save As, Print, Close, and Exit. Separators follow the Open, Save As, and Print options.

Windows Only: If the File menu includes any of the following options, they are ordered: New, Open, Close, Save, Save As, Print, Print Setup, and Exit. Separators follow the Save As and Print Setup options.

Windows Only: If the File menu includes a list of most recently used files, the list precedes the Exit option. Selecting an option with a file name opens a window containing the file. If the file is already open, selecting the option raises that window to the front. The number of files in the list can range from three to eight but remains constant within the application. When a file is opened, the file name is placed at the top of the list in the menu and given the number 1 (e.g., 1 TEST.DOC) which serves as its mnemonic. When another file is opened, it is added to the top of the list (and given the number 1) and the previously opened files are renumbered and move down in the list.

Edit menu. If an Edit menu is present, it contains options that enable users to modify the data in the window. If File and Edit are both included, they are next to each other in the menu bar.

Motif Only: If the Edit menu includes any of the following options, they are ordered: Undo, Cut, Copy, Copy Link, Paste, Paste Link, Clear, Delete, Select All, Deselect All, Select Pasted, Reselect, Promote. Separators follow the Undo, Paste Link, and Delete options.

Windows Only: If the Edit menu includes any of the following options, they are ordered: Undo, Repeat, Cut, Copy, Paste, Paste Special, Clear, Delete, Select All, Find, Replace, and Links. Separators follow the Repeat, Select All, and Replace options.

View menu. If a View menu is present, it contains options for changing the user's view of the data but does not actually change the data. This menu can also contain options for controlling the display of interface elements such as toolbars.

Options menu. If an Options menu is present, it contains options for customizing the application.

Window menu (Windows Only). If a Window menu is present, it contains options for manipulating document windows. The menu includes New Window and window arrangement options (Tile, Cascade, or Arrange commands) and a list of open windows (similar to the list of most recently used files in the File menu). Selecting a window name from the menu raises that document window to the front and gives it focus. The active window is indicated by a check mark preceding the window name

in the menu. The list can contain up to nine window names; if more than nine windows are open, the list includes a More option that, when selected, displays a dialog window with the names of all open document windows from which users can select.

Help menu. The Help menu provides access to additional information about the window or the application. See section 11.4 for additional information on application-level help.

Motif Only: If the Help menu includes any of the following options, they are ordered: Overview, Index, Table of Contents, Tasks, Reference, Tutorial, Keyboard, Mouse, Mouse and Keyboard, On Item, Using Help, and About.[2] The minimum set of options included in the menu is Overview, Tasks, Reference, On Item, Using Help, and About <application>. In addition, if the application provides unique function keys or accelerators, a Keyboard option is also included in the menu.

Windows Only: If the Help menu includes any of the following options, they are ordered: Contents, Search for Help On, Tutorial, How to Use Help, and About.

8.1.2.4 Arrangement of Controls

The main area of a window is organized into subareas based on the nature of information being presented or the type of action required by users.[3] Controls that perform similar or related functions are grouped together and surrounded by a frame, as shown in figure 8-3.

Motif Only: The frame has an "etched in" effect in order to be clearly different in appearance from a window control; a frame does not have a "shadow in" or "shadow out" effect since these effects are used for window controls (e.g., a text field or a push button). Frames include a margin (i.e., have a non-zero value for margin height and width) in order to provide sufficient space between the frame and the objects within it (per Kobara).

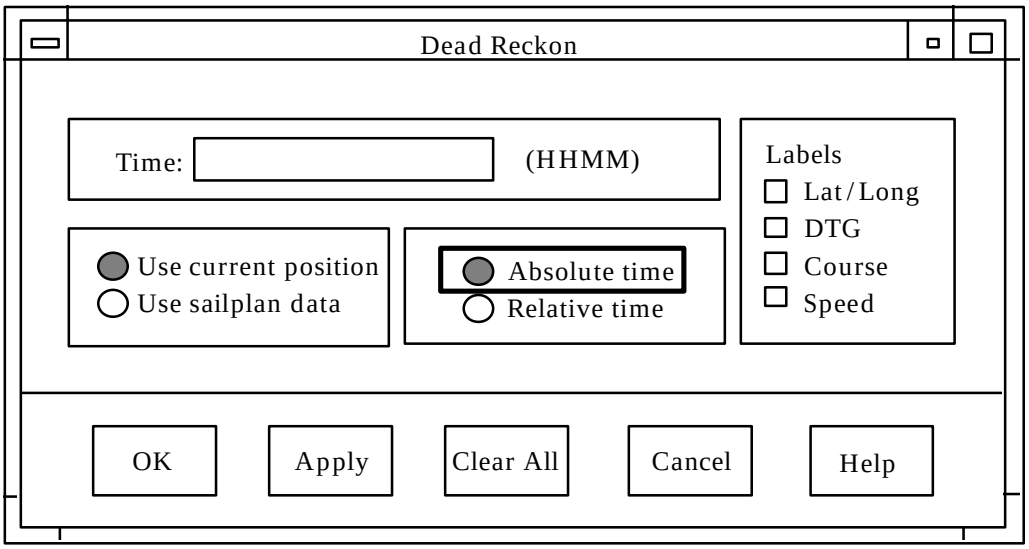


Figure 8-3. Example arrangement of control groups in a Motif window.

[2] These Help menu options supercede the two help models available in previous versions of Motif.

[3] The design of expandable windows will be addressed in a future version of this document.

If a group of controls includes a heading, it is a label describing the function performed by the controls. The heading is placed either inside the frame or in the frame. If the heading is longer than the text in the controls, the size of the frame is extended so that it is wider than the heading. The heading is either left justified or centered within the frame; it is presented in mixed case, following normal capitalization rules, and is not followed by a colon.

The controls in a group are arranged in one or more rows or columns within the frame. The preferred orientation for a group of radio or check buttons is vertical and left-aligned; if placed horizontally, space is sufficient (at least twice the distance between the button and its label) so the button is paired with the label on the right, not left.

Motif Only: Groups of radio and check buttons use the default dimensions defined by Motif, including the value for spacing between the indicator and label.

When a window is initially displayed, all of the controls in the window reflect the current state of the application. For example, a window that allows users to change the font size of text in the application is displayed with the current font selected. If there is a preferred or expected choice within a window, a default option is defined within a control (e.g., a list box) or group of related controls (e.g., a set of check buttons), and this choice is selected (i.e., highlighted) when the window is displayed. If the expected choice cannot be anticipated, then a default is not designated. Controls that become unavailable are dimmed and not available for selection. Controls that are never available to users (e.g., if access to them is password controlled) do not appear in the window. When one of the controls in a window is selected but not executed (e.g., users choose the Cancel push button in the window), the selection is not saved and the control reverts to its original state when the window was first displayed.

8.1.2.5 Availability of Scroll Bars and Window Panes

If a scrollable area in a window is too small to view all of the contents, scroll bars are provided to allow scrolling of the area. The scroll bars are located to the right or at the bottom of the area. When scroll bars are provided, they scroll the main part of window only and not the menu bar or message bar in the window.

Motif Only: A scrollable window always displays its scroll bars regardless of whether the viewable window is the same size or smaller than the underlying data. The scroll bars are positioned 4 pixels from the viewable area in a scrolled window (per Kobara).

Windows Only: Scroll bar(s) remain displayed even if they become inactive.

Window panes are used to separate control areas when space is limited or to present two simultaneous views of the same data in a single window. Panes can be arranged either horizontally or vertically in a window.

Motif Only: Separators and sashes serve as the boundary between panes. Users resize panes by dragging the boundary between the panes using BSelect or BTransfer or by moving the sash using the arrow keys; making one pane larger makes the other pane smaller but does not affect the overall size of the window. Per Kobara, spacing in a paned window is set to the default of 8 pixels so that the sash and separator between panes remain displayed. In addition, sash height is set to 6 pixels, and sash width is either the same as scroll bar width or set to 12 pixels if no scroll bars are used. Finally, the distance the sash is indented is the same as the dimension by which the vertical scroll bars are offset from the right edge of the window; if there are no scroll bars, sash indent is set to -4, which places the sash close to the right edge of the pane.

Windows Only: Split boxes are used to divide the window into panes and then adjust the size of the viewing areas. A split box is a solid box located at the top of a vertical scroll bar or at the

left end of a horizontal scroll bar. Users can drag the split box to divide the window into separate panes; double clicking on the split box divides the window in the middle. A split bar serves as the boundary between the panes. Dragging the split box or split bar to either end of the window closes the pane in the direction of the drag. When a window is split, scroll bars are displayed (along with the split bar) so that users can scroll each pane (i.e., perpendicular to the direction of the split) independently within the window.

8.1.2.6 Arrangement of Push Buttons

Motif Only: Push buttons are displayed horizontally, centered at bottom of the window, and separated from other controls with a separator, as shown in figure 8-3. Push buttons are ordered from left to right based on the sequence in which they will be used, with the most frequently used button on the left. Buttons indicating positive actions are at the left, followed by buttons indicating negative actions and canceling actions. A Help push button is included in every window and is the rightmost button.

Windows Only: Push buttons can be placed along the right margin or across the bottom of a window; the preferred implementation in the DII is the latter. Buttons that initiate actions (e.g., Find, Save) are placed at the left, followed by other push buttons (e.g., "GoTo," "GoSub"), and then Help. If there is an OK button, it is placed first (even if it is not the default) and followed by Cancel, with both separated from the other action buttons. If there is no OK button, Cancel follows the other action buttons but precedes the "GoTo," "GoSub," and Help buttons.

Push buttons appear in the same order throughout the application. A window contains no more than seven push buttons, including Help. Close and Cancel are not included as push buttons in the same window.

Windows Only: Windows that support multiple actions contain a Cancel push button when initially displayed; if the actions performed in the window make irreversible changes to data, the label of the Cancel button changes to Close as soon as the first such action is executed. When the window is closed and then reopened, the button label reverts to Cancel.

If users can perform multiple actions within a window and these actions can affect different objects or data elements in the window, push buttons are labeled to reflect the object(s) that each button affects and are located near the object(s) to which they relate. Push buttons related to overall window functionality (e.g., OK, Cancel) are placed along the bottom of the window.

Motif Only: If users can perform mutually exclusive actions (e.g., Pause and Resume) in a window, the window contains a push button for each action, and the label of the button that is not available at the time is grayed out. The window does not contain a single push button whose label changes to indicate the action available.

Windows Only: If the function performed by a push button changes depending on the state of the application, the label changes accordingly. A single push button, rather than separate buttons, is used in the window.

Windows Only: When a push button with an ellipsis in its label is selected, a dialog window requesting additional information is displayed. The parent window containing the button can close when the child window opens (i.e., a "GoTo" window) or the parent window can remain open (i.e., a "GoSub" window). In both cases, if the child window can also be opened as a result of a menu selection, the window title is the same as the menu option. If users open a "GoSub" window, make changes in the window that cannot be undone, and then return to the parent window, selecting Cancel in the parent cancels all changes made in both the parent and child windows.

8.1.2.7 Default Push Buttons

If a default action is available, it is the action that users are most likely to execute in the window. If an expected choice cannot be anticipated, there is no default defined for the window; <Enter> (or <Return> in Motif) has no effect, and users must select one of the available push buttons to execute an action.

> Motif Only: When a window is initially displayed, the default push button is the leftmost button in a group of push buttons.

The same button is the default whenever the window is displayed. The default designation may assigned to a different push button depending on the control that has keyboard focus in the window. When keyboard focus is on a push button, its action is the default, and it is shown with default highlighting. This highlight moves with the location cursor during keyboard navigation in a group of push buttons and returns to the original button when focus leaves the push button group. If the default action in a window varies, one push button always shows the default highlighting except when there is no default action currently available. If focus is outside a window, the highlight is displayed on the push button whose action will be the default when focus returns to the window.

When more than one action is available in a window, the default push button is the nondestructive one. If the default highlight moves to a push button other than the one originally identified as the default, the highlight reverts to the original button when the window is closed and then opened again. The action performed by the default push button is reversible (e.g., by selecting an Undo menu option or a Cancel push button).

8.1.2.8 Tool Bars

If a tool bar is included in a window, it provides redundant access to functionality available elsewhere in the window (e.g., in a pull-down menu). For example, the buttons in a tool bar can invoke modes (e.g., different drawing tools), apply settings (e.g., select font style or size), or execute actions (e.g., invoke a print command).

> Motif Only: A tool bar is used only in windows with a menu bar; if present, the tool bar is located at the top of the window, below the menu bar, as shown in figure 8-4. The tool bar is the same width as the window and the same height as the menu bar.
>
> Windows Only: A tool bar can occupy a fixed position within a window (e.g., below the menu bar, along the left margin or bottom of the window), or it can be placed in a separate dialog window and be movable. In the latter case, the tool bar is always displayed in front of the window to which it applies. The dialog window with the tool bar includes a title bar (so the window can be dragged to a new location) and a Window menu with Move and Close options.

A window with a tool bar also includes a message bar at the bottom of the window so that information about the purpose of a button in the tool bar can be provided when the pointer is on the button or the button has focus. Users have the option to hide or show the tool bar (e.g., from a toggle-type option in the View menu).

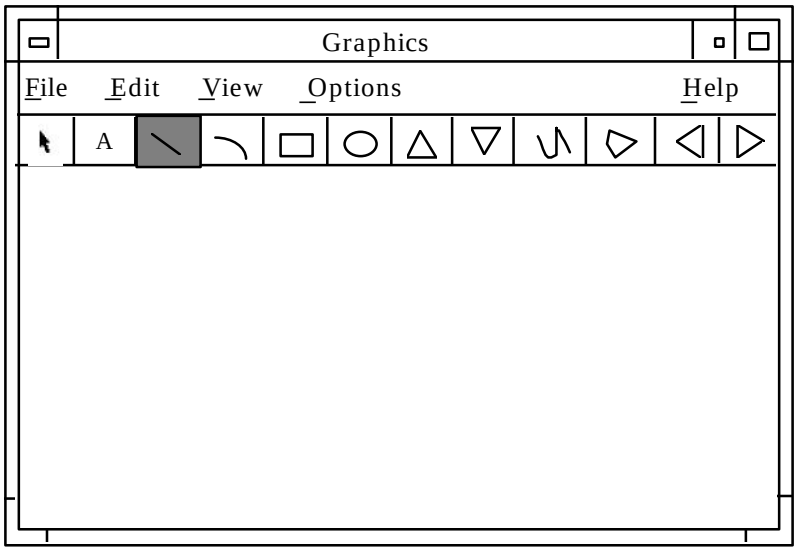


Figure 8-4. Example tool bar in Motif.

Motif Only: A tool bar contains no more than 20 action buttons. The buttons are of equal size and evenly spaced across the tool bar. They are arranged in an order expected by users; if one does not exist, they are arranged by frequency of occurrence, sequence of use, or importance.

Windows Only: A toolbar can contain drop-down lists as well as groups of action buttons; spacing is 12 pixels between the lists and buttons and 6 pixels between button groups. A toolbox (i.e., a toolbar placed along the left margin of a window) contains groups of action buttons arranged vertically, with no space between the buttons.

The normal appearance of a button is raised; when selected, the button is recessed and changes appearance (e.g., highlights) to indicate its selected state. A button that becomes unavailable changes appearance (e.g., is grayed out) to show that it cannot be selected. Whenever a menu option becomes unavailable, the corresponding button in the tool bar is also shown as unavailable.

The labels for buttons in a control bar should be presented as icons, with graphics that are the same size in each button.

Motif Only: Button size is 24 x 24 pixels.

Windows Only: Normal button size is 24 x 22 pixels, with a graphic image 16 x 15 pixels. If space is limited, button size is 22 x 18 pixels, with a graphic image 16 x 12 pixels. The graphic image is a black outline, with white fill within the outline where needed.

The icon represents a verb (rather than a noun) and can depict a before and after representation of the action (e.g., a small and large version of an object, connected by an arrow, to indicate Magnify), the tool that would accomplish the action (e.g., a pair of scissors to indicate Cut), or the action itself (e.g., a paintbrush filling an object with color to indicate Paint). One of these schemes is selected, and all of the icons in a tool bar designed to fit this scheme. Text labels describing the action executed by the button may also be included with the icon graphics.

Windows Only: If text is included with the graphic, the size of the button is larger and dependent on text length. The graphic is placed either above the text or to the left of the text; in the latter case, the button has the same height as a normal push button (i.e., 22 pixels), with 4 pixels between the graphic and text.

When a button that invokes a mode is selected, the button remains selected (i.e., highlighted) as long as the mode is in effect. The pointer shape changes to indicate type of operation users can perform while in the mode. The pointer has this shape whenever it is in the window where the mode is in effect. If users move the pointer outside this window, the pointer changes to the appropriate shape. A tool bar either provides a button for returning to an unselected state (i.e., exiting the mode) or automatically returns to an unselected state after an action is executed.

8.1.2.9 Message Bar

If a window includes a message bar, it is used to provide noncritical application messages to users, present simple help, or indicate status pertaining to the window. When the message bar is used to indicate status, a progress message is displayed when the action is initiated (e.g., "Drawing map...") and updated when the action is completed (e.g., "Drawing map...Done"). The text is removed from the message bar within 5 sec of action completion.

A message bar displays read-only text; users cannot type or modify any information presented in this area.

Motif Only: The message bar can have the same appearance as a noneditable text area (as was recommended in previous versions of Motif); the preferred implementation indicated in CDE is to widen the margin area at the bottom of the window so that messages can be presented there.

Windows Only: A status bar is a more elaborate version of a message bar in which messages as well as information about the current state of the application (e.g., cursor location, modes in effect) are displayed. If a status bar is used in a window, the application provides users with a way to hide or show the bar (e.g., as an option in the View menu) as desired.

8.1.2.10 Draggable Objects in Windows (Motif Only)

As indicated in section 3.5.1.1, the application is to provide a drag and drop capability for all objects represented as icons and for all window elements that users can directly manipulate. This capability provides redundant access to functionality available elsewhere (e.g., buttons, menus, dialog windows) in the application.

An icon graphic is included in a window to indicate that it contains a draggable object. The graphic is the same as the one used to represent the object in the File Manager on the desktop. The icon is placed next to any display of the contents of the object if one is present in the window. If there is no such display, the icon is placed in the upper right corner of the window. The icon is 32 x 32 pixels and includes a label describing the kind of object the icon graphic represents. The graphic is also used as the source indicator in the drag icon.[4]

8.1.2.11 Pop-up Menus and Text Fields in Windows

Pop-up menus are provided for those functions or objects in a window for which redundant access would improve task performance. The availability of these menus is limited to frequently executed functions or frequently selected objects included in a window. For example, pop-up menus with edit commands can be provided for the text fields in a window so that users do not have to move the pointer to and from an Edit pull-down menu. Similarly, a pop-up menu containing frequently executed actions

[4] The implementation of drag and drop for attachments will be addressed in a future version of this document.

can be available in a window so that users do not have to move the pointer to and from the push button area of the window.

Windows Only: Read-only pop-up text fields are used to display additional information about the text in a field when space within a window is limited. The presence of a pop-up field is indicated by underlining the word(s) in the text to which it relates. Clicking BSelect on the underlined text displays the pop-up field; clicking anywhere outside the field dismisses it. When the field is displayed, it is placed so that its top left corner is at the same position as the top left corner of the original text.

8.1.2.12 Mnemonics and Accelerators in Windows

Windows Only: Mnemonics are available as an additional method for keyboard navigation among controls in a window. When mnemonics are implemented, they behave as indicated in section 3.4.2 and 5.5 and use the characters listed in appendix C. The OK and Cancel push buttons do not have mnemonics, given that <Enter> activates the OK button if it is the default and <Esc> activates the Cancel button.

The use of accelerators in windows is limited to the set of function keys listed in appendix A. Accelerators are not assigned to individual controls since this functionality would not be visible in the window (i.e., part of the label for the control).

8.1.2.13 Document Windows (Windows Only)

A document window has the same window components as a primary window, and its Window menu contains the same options as the Window menu in the application window. The title of a document window is the name of the document and presented in mixed case.

Document windows appear within the borders of the application window. When a document window is maximized, the window is closed, and the data from the window are displayed in the application window. The title of the application window changes to include the document name (i.e., application name, followed by a hyphen, and then the document name), the Window menu button from the document window is displayed at the left end of the menu bar in the application window, and a Restore button for the document window is added to the right end of the menu bar. In addition, the application window has a scroll bar if the document window requires scrolling.

If the application has a menu bar, it appears in the application window, along with any controls (e.g., tool bars) that apply to all document windows. In applications that support different document types, the content of the menu bar can change depending on the document that currently has focus. Menu titles can be temporarily removed from the menu bar if no documents are open; the only menu titles included in the menu bar are those that can be executed when no documents are open.

8.2 DIALOG WINDOWS

8.2.1 Window Components

A dialog window contains a frame, a Window menu button, and a title bar; the window does not have resize borders or Minimize or Maximize buttons. The Window menu in a dialog window includes (in this order) Move and Close options.

Windows Only: If a dialog window is not movable, it does not have a title bar, and Move is not an option in the Window menu.

The area inside the window frame of the window contains a title and a main area. The main area includes a control area for presenting messages or controls and a push button action area at the bottom of the window for executing actions.[5]

Motif Only: Message windows are modeless whenever possible.[6] An error message window is modal only if it is critical that users acknowledge having read the message prior to continuing to interact with the application.

Windows Only: Message windows are modal. Critical message windows are system modal while information and warning message windows are application modal. The frame of a modal window has a colored inner border; the frame of a modeless window does not.

8.2.2 Window Design Guidelines

Motif Only: The title of a dialog window includes the name of the application and describes the purpose of the dialog window. The window contains a separator between the control area and action area of the window.

Windows Only: The title of a message window is the application name and does not include the word "Error."

If a dialog window includes a text message, it uses language that is meaningful to users and requires no further documentation or translation. The text is left justified within the window. When a message contains more than one sentence, the important information is placed at the start of the message. The text is worded so that the action users are asked to perform can appear as a push button in the window. For example, a window displaying the message "Confirm deletion of file" contains Delete and Cancel push buttons. The application does not present timed-information windows (i.e., message windows that present information for a fixed time period) and then resume processing without requiring a user response.

A dialog window contains at least one push button that either performs the dialog window action and dismisses it (e.g., OK) or dismisses the window without taking any action (e.g., Cancel). A default push button is available in each dialog window in the application. If an action executed in a dialog window results in an error that generates an error message window, the dialog window remains displayed while the error window is presented and then dismissed.

Push button order is OK / Cancel / Help in modal dialog windows, with OK designated as the default. Push button order is OK / Apply / Cancel / Help or OK / Apply / Reset / Cancel / Help in modeless windows; OK is the default in windows that perform single actions, and Apply the default in windows that perform multiple actions. <Cancel> has the same effect as selecting the Cancel push button in the window.

When a dialog window containing critical information is displayed, it is accompanied by auditory feedback (e.g., a beep) as a secondary indicator to attract the user's attention. Users can set auditory signals at a very low intensity or disable them as required (e.g., for rig-for-quiet operations on submarines).

---

[5] Expandable windows and Property and About dialog windows will be addressed in a future version of this document.

[6] Previous versions of Motif recommended that error, question, and warning message windows be application modal and information and working message windows be modeless. CDE recommends that all message windows be modeless whenever possible.

8.2.3 Message Dialogs (Motif Only)

8.2.3.1 Error Message Windows

An error message window, shown in figure 8-5, is displayed to inform users when an error has occurred. The window contains the error symbol, a text message, and the following push buttons (in the order indicated):

OK, Help

Continue, Cancel, Help.

The text message describes the error, why it happened, and what should be done to correct it.

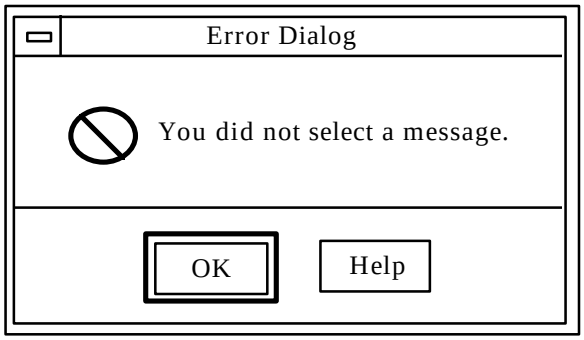


Figure 8-5. Example error message window in Motif.

8.2.3.2 Information Message Windows

An information message window, shown in figure 8-6, is displayed to convey noncritical information that requires acknowledgment by users (the message bar of a window is used for messages that require no acknowledgment). The window contains the information symbol, a text message, and the following push buttons (in the order indicated):

OK

OK, Help.

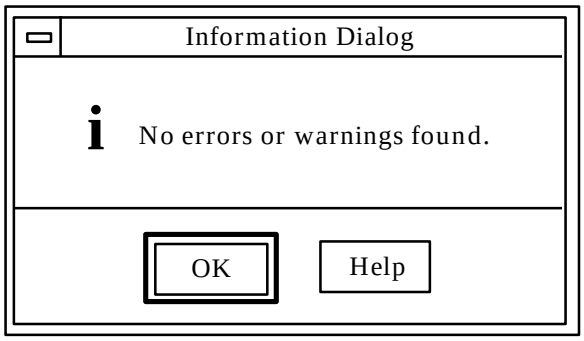


Figure 8-6. Example information message window in Motif.

8.2.3.3 Question Message Windows

A question message window, shown in figure 8-7, is displayed to request clarification of a previous response. The window contains the question symbol, a text message, and the following push buttons (in the order indicated):

Yes, No, Help.

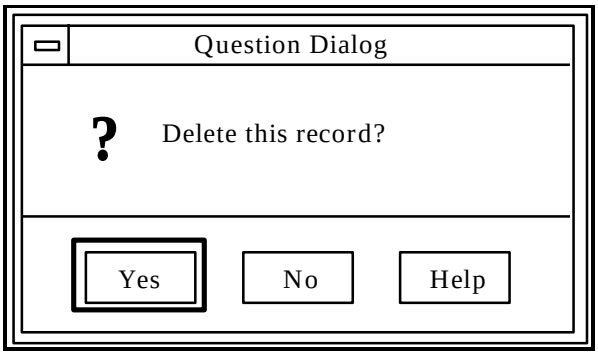


Figure 8-7. Example question message window in Motif.

8.2.3.4 Warning Message Windows

A warning message window, shown in figure 8-8, is displayed to present critical messages concerning the consequences of an action and to allow users to cancel a destructive action. The window contains the warning symbol, a text message, and the following push buttons (in the order indicated):

Yes, No, Help

Continue, Cancel, Help.

An audio signal accompanies the window to alert users to the warning.

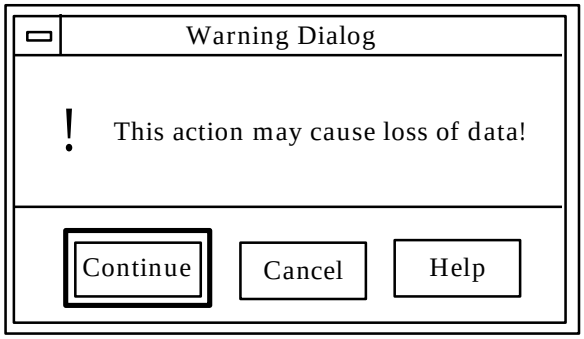


Figure 8-8. Example warning message window in Motif.

8.2.3.5 Working Message Windows

A working message window, shown in figure 8-9, is displayed when the processing time in response to a user's request exceeds 10 sec or when a user may want to cancel the operation that is in progress. The window contains the working symbol, a text message, and the following push buttons (in the order indicated):

OK, Help

OK, Cancel, Help

OK, Stop, Help

OK, Pause, Resume, Stop, Help.

Cancel interrupts the operation and returns the application and data to its state before the operation was activated. If a return to that state is not possible, Stop is used instead of Cancel. Stop interrupts the operation but does not reverse any changes already caused by the operation.

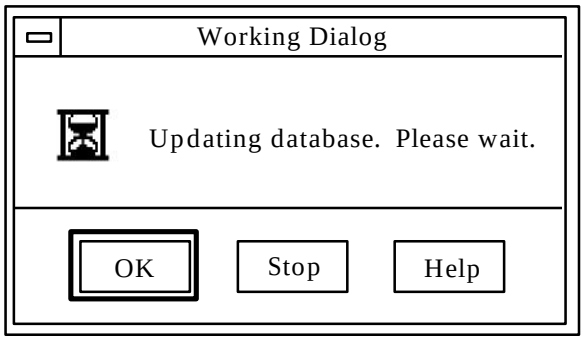


Figure 8-9. Example working message window in Motif.

During lengthy processing (in excess of one minute), a working window is updated (e.g., the text is changed or the trough in a gauge is filled) to indicate the status of processing if update information is available. The window remains displayed until the action is complete, the window doing the processing is minimized, or the user selects Cancel. When processing is complete, the window is removed (without user action). Users can cancel the operation in progress if desired, and they have to confirm the action before it is executed if unsaved data will be lost.

8.2.4 Message Dialogs (Windows Only)

8.2.4.1 Information Message Windows

An information message window, shown in figure 8-10, is displayed to provide information about the results of commands. The window contains the information symbol, a text message, and the following push buttons (in the order indicated):

OK

OK, Help.

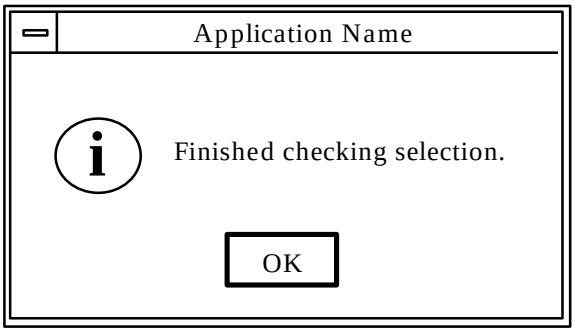


Figure 8-10. Example information message window in Windows.

8.2.4.2 Warning Message Windows

A warning message window, shown in figure 8-11, is displayed to present error information or to allow users to cancel a destructive action. The window contains the warning symbol, a text message, and a push button for each choice available in the window plus a Help button. The message text may be worded as a question; if so, the window includes the following push buttons (in the order indicated):

Yes, No, Help.

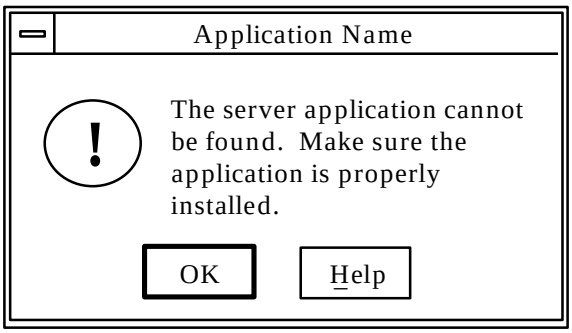


Figure 8-11. Example warning message window in Windows.

8.2.4.3 Critical Message Windows

A critical message window, shown in figure 8-12, is displayed to present messages that must be corrected before users can continue to work in the application. The window contains the critical symbol, a text message, and push buttons for each choice available in the window plus Help.

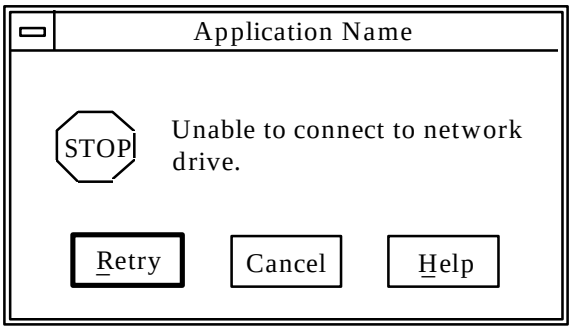


Figure 8-12. Example critical message window in Windows.

8.2.5 Selection Dialogs (Motif Only)

8.2.5.1 Command Windows

A command window is displayed when users need to enter keyboard commands. The window, shown in figure 8-13, contains a list box that displays a command history and a text field for entering commands; the window does not include any push buttons. The list has a vertical scroll bar when the command history exceeds the visible area in the list. The command history is cleared whenever the application is exited, and resumed when the application is launched again. The text field is wide enough for users to view and read an entire command; a horizontal scroll bar is not included unless command lines are unusually long.

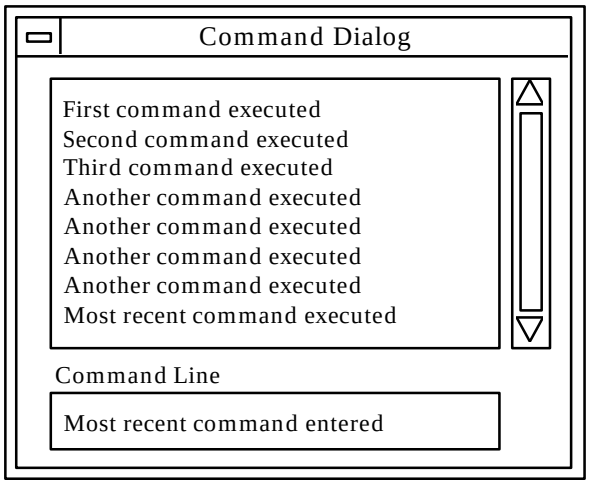


Figure 8-13. Example command window in Motif.

Selecting an item from the list displays it in the text field. <Enter> or <Return> executes the command and adds it to the bottom of the command history list. <Tab> moves the location cursor between the list and the text field. When focus is on the text field, <Up>, <Down>, <Ctrl><Home>, and <Ctrl><End> move the location cursor among items in the list and change the contents of the text field.

8.2.5.2 Prompt Windows

A prompt window, shown in figure 8-14, is displayed to request information needed to continue processing. A prompt window includes a message stating what information is needed, a text field for typing, and the following push buttons (in the order indicated):

OK, Cancel, Help
OK, Apply, Cancel, Help
OK, Apply, Reset, Cancel, Help

The text field has keyboard focus when the window is initially displayed.

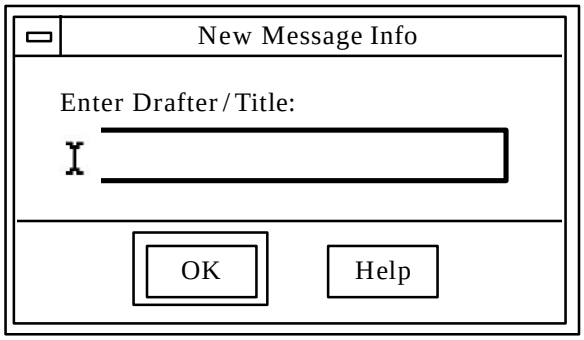


Figure 8-14. Example prompt window in Motif.

8.2.5.3 Selection Windows

A selection window, shown in figure 8-15, is displayed when users need to make a selection from a list of choices. The window includes a list box containing the choices available, a text field for displaying and editing the choice, and the following push buttons (in the order indicated):

OK, Cancel, Help
OK, Apply, Cancel, Help.

Both the list box and the text field include a heading that describes their contents. The list has a vertical scroll bar when the number of items exceeds the visible area in the list.

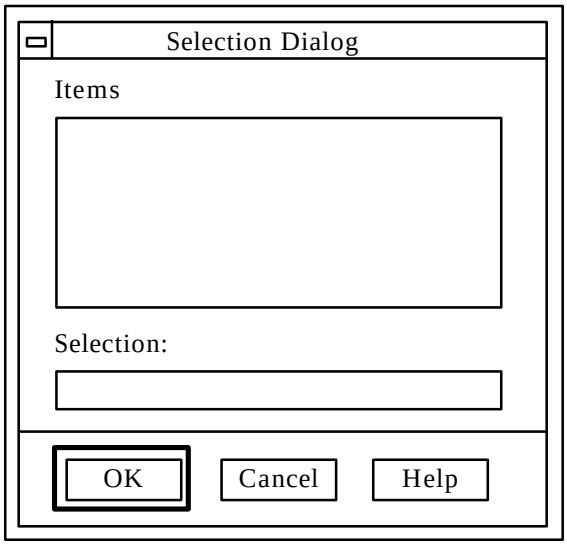


Figure 8-15. Example selection window in Motif.

Selecting an item from the list box displays it in the text field. If users type in the text field, the list scrolls to that item in the list. If the text typed in the text field does not match any items in the list, users are prompted to add the item to the list. When users select OK or press <Enter> or <Return>, the selection is executed and the window closed. <Tab> moves the location cursor between the list and the text field. When focus is on the text field, <Up>, <Down>, <Ctrl><Home>, and <Ctrl><End> move the location cursor among items in the list and change the contents of the text field.

8.2.5.4 File Selection Windows

A file selection window, shown in figure 8-16, is displayed when users need to choose a file or directory.[7] A file selection window contains a text field for displaying and editing the current directory path, list boxes for displaying directory and file names, a text field for displaying and editing a file name, and OK/Update/Cancel/Help push buttons. The text fields and lists are labeled as in figure 8-16. The File text field can be omitted if the window is used to choose an existing file or directory. The OK push button can be replaced with a command that matches the action for which the window was displayed. When a file selection window is used to specify an existing file, the command is Open and it is the default action in the window. When the window is used to specify a new file name, the command is Save and it is the default action.

[7] CDE provides a file selection window with additional features beyond what was available in previous versions of Motif.

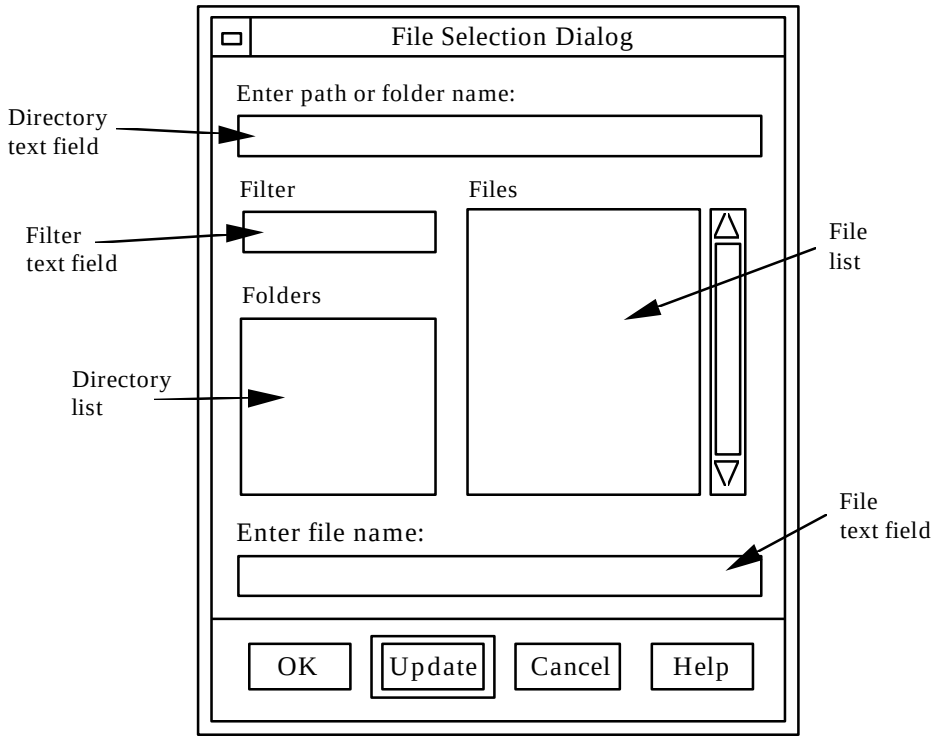


Figure 8-16. Example file selection window in Motif.

The items in the Directory and File lists are presented in alphabetical order. The first item in the Directory list is the parent directory and labeled "..". The file selection window does not display hidden (i.e., dot) directories or files unless users need access to these types of files; if access is required, the window includes a check button that allows users to show or hide these files. The Directory text field presents the full path name; the File text field and the lists in the window show relative path names.

When users open the file selection window associated with a particular primary window, the directory location displayed is the default for that primary window. If users change the directory and then reopen the file selection window, the directory location is the one that was previously set by the user. When users close the primary window, the directory location in the file selection window reverts to the default for the primary window. If the application supports multiple primary windows, the directory reverts to the default defined for that primary window. The default is selected to match the task being performed in the window. For example, when the user executes an action to save a file, the file selection window contains the user's home directory.

When users open a file selection window, the File text field has keyboard focus, and the File list displays the contents of the current directory. The File list is updated when users edit the Directory text field and press <Enter> or <Return> or when they select a directory in the File list. When users select a file from the File list, the file name appears in the File text field. The application executes the selection(s) in the window when users select an item in the File list and activate the OK (or comparable) push button, when users double click BSelect on an item in the File list, or when users select a file name and press <Return> or <Enter> when the File text field has keyboard focus. Users are prompted to confirm the action executed if it will overwrite an existing file.

8.2.5.5 Print Windows

A print window, shown in figure 8-17, is displayed when users need to select options for printing a file, a selection, or other type of object. The window contains a common area with standard information

about the print job, an optional area with information specific to the application or function, and Print / Cancel / Help push buttons. The common area, located in the top part of the window, displays the name of the file or object type and includes controls for entering the name of the printer destination, the number of copies desired, and the text to appear on the banner page, if any. A text field or combo box can be used to enter the printer destination. The default entry in this field is "Default" (i.e., the printer that is the default destination in the system); users can select or type any other valid printer name. The application saves the last user entry in this field and displays it when the window is opened again. A text field or spin button can be used to enter the number of copies, and a text field can be used to enter the banner page title.

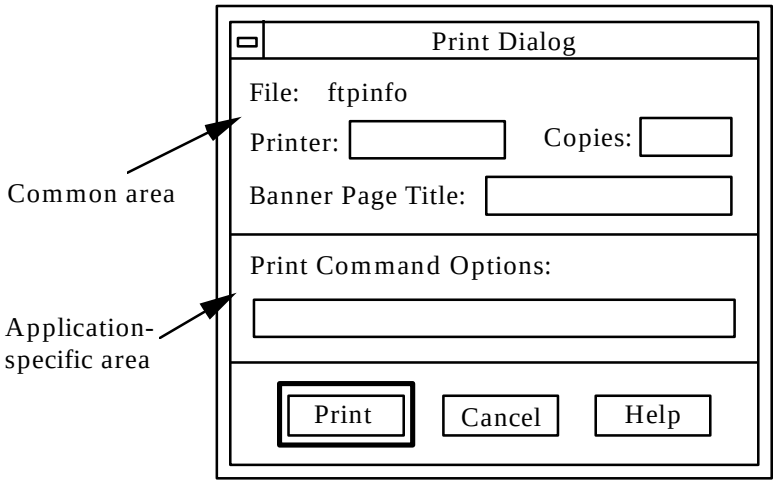


Figure 8-17. Example print window in Motif.

The same information is presented in the common area in all print dialogs, with any application-specific controls placed in the optional area in the lower part of the window. These controls request users to enter information such as Page Range, Priority, Orientation, Resolution, Paper Size, and Paper Source. If Reset and Print Preview actions are included in the window, they are available as push buttons and inserted between the Print and Cancel buttons. Separators are used between the common area, the optional area, and the push buttons.

8.2.6 Selection Dialogs (Windows Only)[8]

8.2.6.1 File Open and File Save Windows

The standard File Open window is used to choose a file or directory. This window contains a noneditable text area displaying the current position in the directory tree, list boxes for displaying directory and file names, a text field for displaying and editing a file name, drop-down lists for selecting drives and file types, and OK and Cancel push buttons. Users navigate in the window by either selecting from the Drives or Directories control or by typing this information (along with a file name) in the File Name text field. When users switch to a new drive, the contents of the Directories list shows the contents of this drive, with the root directory at the top. Users can select a file by either typing in the File Name text field or selecting one of the items in the list box below the text field. If desired, users can filter the types of files included in this list by selecting a file type from the List Files of Type drop-down list.

[8] Other common selection dialogs in Windows will be addressed in a future version of this document.

A File Save As window has the same basic format as a File Open window, except the file names shown in the list box are grayed out (i.e., unavailable for selection) and the List Files of Type drop-down list is relabeled Save File as Type, with the type selected indicating the format of the file to be saved.

8.2.6.2 Print Windows

The standard Print window is used to submit a print job. This window identifies the printer to which the job will be sent and contains controls for specifying the print range, print quality, and number of copies and choosing to print to file or to collate the copies. These controls appear in the common part of the Print window, with application-specific controls available in an optional area in the lower part of the window.

8.3 CONSIDERATIONS IN WINDOW DESIGN

The specifications in this section are provided to assist developers in making effective design decisions regarding the windows in their applications. A key assumption underlying these specifications is that effective design begins with an understanding of the users' perspective and how the application will support the tasks they perform. Note: These specifications are not included in the UIS checklist.

8.3.1 Selecting Controls to Match User Actions

The controls that appear in a window match the actions that users are expected to execute in the window. The following are guidelines on how to choose the most appropriate controls:

Use radio buttons, an option menu, or a list box when users need to select from discrete values or choices; use a scale or spin button when users need to select from a continuous range of values.

Use a group of radio buttons when it is important for users to see all of the settings available in a group; use an option menu or drop-down list box when users need to see only the current choice.

Use radio buttons or an option menu when the set of options from which to choose is not likely to change; use a list box when the options from which to choose may change.

When users have to make a single selection, use a group of radio buttons if there are up to 5-6 choices; use an option menu if there are up to 10-12 choices; and use a list box if there are more than 12 choices.

When users have to make multiple selections, use check buttons if there are up to 7 choices; use a list box otherwise.

Use push buttons for frequently executed actions when space is available to display the buttons; use tear-off menus for these actions if space is limited.

Use option menus for setting values or choosing from a set of related options; use push buttons for activating commands.

8.3.2 Arranging Controls by Importance and Scanning Order

The most important information and controls associated with the task being performed in a window is located in the upper left part of the window, working down to the less important information at the bottom of the window. In addition, information in a window is arranged to accommodate the

possibility that users will resize the window. Placing the most important objects in the upper-left corner of the window allows users to manipulate these objects even if the window is reduced to a size that is smaller than normal. Finally, if a window contains task-critical information, it is visually set apart from other information in the window so that it can be easily seen. The minimum separation is one character space above and below and two character spaces before and after the critical information.

Windows are designed according to users' natural scanning order and probable selection sequences. In most cases, this order is from left to right and top to bottom. For example, when presenting a series of radio or check buttons, the most frequently used button is the top (or leftmost) button in the group. The window is designed from the perspective of what is logical to users and appropriate to the actions being executed, not what is logical or appealing to developers.

8.3.3 Designing for Efficiency in Task Performance

The controls in a window are arranged so that users can move quickly and easily among them. The amount of pointer movement and/or the number of keystrokes required to perform a task is minimized. Likewise, objects are arranged to minimize the amount of hand movement between the keyboard and pointing device when users work in a window; users are not forced to change input devices when performing a task.

The number of actions users have to execute in order to complete a task is minimized. For example, the application prefills text entry fields (e.g., current date, ownship name, position) with default data values whenever possible. Similarly, if the application repeats the same data fields in multiple windows, users have to fill the field once, with the application automatically placing this value in the field as the default whenever it recurs.

Developers need to adopt a consistent organizational scheme for the key elements in a window and then apply that basic scheme to all windows in the application. The same window design is employed whenever users have to perform the same basic task; for example, a single design is used to present identifying information on data records, whether the data are tracks or messages. Different or distinctive elements can appear in a window to fit the task being performed, but these elements are consistent across windows within the application.

The number of intermediate windows that users must interact with in order to complete a task is minimized. The application is designed so that users are able to perform each major task in a single window containing all of the information relevant to the task. Users can complete the task without having to refer to information not included in the window. Developers need to be sensitive to problems created by presenting users with a sequence of independent windows, each of which contains only a portion of the overall task being performed. While this type of design may be efficient from a software development perspective (e.g., by allowing re-use of previously designed windows), it can produce an unnecessarily complex task navigation problem for users, increase the difficulty of the task being performed, and place an unnecessary memory load on them to remember information not included in the current window.

8.3.4 Minimizing the Opportunity for User Error

The application is designed to minimize the opportunity for user error. Users can perform only legal operations, rather than allowed to execute an incorrect operation and then informed that they have made an error. Only those actions that are relevant are available for user selection. Controls that cannot be selected are both visually deemphasized by graying out to indicate their unavailability as well as disabled (so that no action is executed if they are selected by users). The application provides visual and behavioral cues to prevent users from performing an illegal or incorrect operation, rather than relying on error messages to inform users after an error has been made.

The actions available for each control displayed in a window match the action(s) users are expected to perform with the control. For example, if users are required to select a single item in a list box, the available selection methods are limited to this type of action. Users are not allowed to select multiple items in the list, after which they are informed in an error message that they should have selected only one item.

Efforts to minimize the opportunity for user error do not constrain users unnecessarily as they work in the application. For example, a window containing several mandatory data fields grays out the OK or Save push buttons until these fields are filled so that users are unable to save the data before completing the mandatory fields. This approach allows users to fill the fields in whatever order they choose or to make corrections as desired before attempting to save the data. A more constraining approach would manage each of the fields so that users have to enter a value before being allowed to leave the field. This approach forces users to perform data entry in a lock-step sequence defined by the application and restricts their ability to perform the task in a manner that does not match this sequence. The application needs to be designed to provide flexibility and support the user's sense of control, while at the same time bounding user interaction to minimize the opportunity for error.